REMARKS/ARGUMENTS

Claims 1-20 are pending in the present application. Claims 2, 5, 12, 18 and 20 have been amended herewith. Reconsideration of the claims is respectfully requested.

I. Objection to Specification

The Examiner objected to the Specification, stating that "filesystem" should be written as two words. Applicants graciously decline such request, as "filesystem" is a commonly known term to those skilled in the art, as evidenced by Attachment A attached hereto. In addition, and as further described below, the Examiner did not interpret "filesystem" to have its plain, ordinary and customary meaning in examining the present application, as required by MPEP 2111.01.

II. Objection to Claims

The Examiner objected to Claim 2, stating "are" should be changed to "is". Applicants have amended Claim 2 accordingly.

III. 35 U.S.C. § 112, Second Paragraph

The Examiner rejected Claim 18 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. This rejection is respectfully traversed.

The Examiner states that "the eighth and ninth instructions" do not have proper antecedent basis. Applicants have amended Claim 18 to depend upon Claim 17 in order to provide proper antecedent basis for such terminology.

Therefore the rejection of Claim 18 under 35 U.S.C. § 112, second paragraph has been overcome.

IV. 35 U.S.C. § 102, Anticipation

The Examiner rejected Claims 1-5, 8-13 and 16-20 under 35 U.S.C. § 102 as being anticipated by Widell et al. (US Patent Application Publication 2005/0055490 A1) ('Widell'). This rejection is respectfully traversed.

For a prior art reference to anticipate in terms of 35 U.S.C. 102, *every element* of the claimed invention must be *identically shown* in a single reference. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990). During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." *In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000). The broadest reasonable interpretation of the claims *must also be consistent with the interpretation that those skilled in the art would reach. In re Cortright*, 165 F.3d 1353, 1359, 49

USPQ2d 1464, 1468 (Fed. Cir. 1999) (emphasis added by Applicants). Applicants will now show that every element of the claimed invention is not identically shown in a single reference.

With respect to Claim 1, it is respectfully submitted that such claim is directed to a system for recovering from a *filesystem* operation failure. The cited reference does not teach any type of filesystem, and therefore does not teach (or otherwise suggest) any type of system for recovering from a filesystem operational failure. It is commonly known to those skilled in the art that a filesystem is a file management system that is used by an operating system to organize and monitor files (see Attachment A; see also Specification page 1, 3rd paragraph). The cited reference does not even use the word 'file' or 'filesystem' in the entire text of such reference, and therefore does not teach any type of file or filesystem. Given the strict identity required of the test for novelty[1], a memory element conflict and associated actions as described by the cited reference does not teach or otherwise describe any type of filesystem[2] or filesystem recovery. Accordingly, as every element of the claimed invention is not identically shown in a single reference - specifically, there is no teaching of a filesystem[3], whereas Claim 1 expressly recites "a filesystem" - Claim 1 has been erroneously rejected under 35 USC 102(e).

Applicants initially traverse the rejection of Claims 2-5 for reasons given above with respect to Claim 1 (of which Claims 2-5 depend upon).

Further with respect to Claim 2, such claim recites "change a second set of data for a second thread, said second thread associated with a second file of said filesystem". As can be seen, this claim recites a *second file* of a filesystem, which is in addition to the first file recited in Claim 1 (and which Claim 2 depends upon). The cited reference does not teach (or otherwise suggest) two files (a first file and a second file), each associated with a separate thread (a first thread and a second thread, respectively). Rather, such reference teaches that either CPUs or threads access a shared memory (page 2, paragraph [0029]; page 3, paragraph [0036]; Figure 1; Figures 3A-3B). A teaching of sharing a common memory, as per the teachings of the cited reference, does not teach two threads each accessing a different file (first file, second file), as claimed in Claim 2. Therefore, as every element of the claimed invention recited in Claim 2 is not identically shown in a single reference, Claim 2 is further shown to have been erroneously rejected under 35 USC 102(e).

---

[1] See, e.g., *Trintec, Indus. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 63 USPQ2d 1597 (Fed. Cir. 2002) where a color *printer* did not anticipate a claim requiring a color *photocopier*.

[2] See, e.g., Attachment A attached hereto.

[3] The cited reference describes recovery from shared memory collisions, and such shared memory is not a filesystem (see, e.g. Widell paragraph [0029], where the shared memory is stated to be a cache line, a variable, or an object in a source language). Distorting the definition of shared meory to also mean a filesystem is not consonant with the art to which the invention pertains.

Applicants traverse the rejection of Claims 8-13 and 16-20 for similar reasons to those given above with respect to Claims 1-5.

Therefore, the rejection of Claims 1-5, 8-13 and 16-20 under 35 U.S.C. § 102(e) has been overcome.

V. 35 U.S.C. § 103, Obviousness

The Examiner rejected Claims 6, 7, 14 and 15 under 35 U.S.C. § 103 as being unpatentable over Widell et al. (US Patent Application Publication 2005/0055490 A1) ('Widell') in view of Carter et al. (US Patent 5,987,506) ('Carter'). This rejection is respectfully traversed for reasons given above with respect to Claim 1.

Further with respect to Claim 7 (and Claim 15), it is urged that the Examiner has failed to provide requisite motivation for combining the references in rejecting Claim 7. As stated by the Federal Circuit, "virtually all [inventions] are combinations of old elements." *Environmental Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 698, 218 USPQ 865, 870 (Fed. Cir. 1983); *see also Richdel, Inc. v. Sunspool Corp.*, 714 F.2d 1573, 1579-80, 219 USPQ 8, 12 (Fed. Cir. 1983) ("Most, if not all, inventions are combinations and mostly of old elements."). Therefore, an examiner may often find every element of a claimed invention in the prior art. If identification of each claimed element in the prior art were sufficient to negate patentability, very few patents would ever issue. Furthermore, rejecting patents solely by finding prior art corollaries for the claimed elements would permit an examiner to use the claimed invention itself as a blueprint for piecing together elements in the prior art to defeat the patentability of the claimed invention. Such an approach would be "an illogical and inappropriate process by which to determine patentability." *Sensonics, Inc. v. Aerosonic Corp.*, 81 F.3d 1566, 1570, 38 USPQ2d 1551, 1554 (Fed. Cir. 1996). To prevent the use of hindsight based on the invention to defeat patentability of the invention, this court requires the examiner to show a motivation to combine the references that create the case of obviousness. In other words, the examiner must show reasons that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed. *In re Rouffet*, 149 F.3d 1350, 47 USPQ 2d 1453 (Fed. Cir. 1998) (emphasis added by Applicants).

In rejecting Claim 7, the Examiner merely states:

> "With respect to Claims 7 and 15, Widell as modified teaches wherein said second file comprises a directory page (Carter, column 30, lines 60-61)."

There is no reasoning given of why a person of ordinary skill in the art, confronted with the same problems as the inventor (maintaining filesystem consistency notwithstanding a filesystem aborted or incomplete operation; Specification page 3, last paragraph), would have been motivated to modify the teachings of Widell to include a directory page as described by Carter. Therefore, the Examiner has failed to provide requisite motivation for combining the references in rejecting Claim 7. Still further, a person of ordinary skill in the art would not have been motivated to make such modification since Widell is directed to shared memory collision processing, and has no concern or regard to accommodating changes made to a directory. Thus, the only motivation for making such a change *must* be coming from Applicants' own description in the Specification, which is improper hindsight analysis.

Therefore, the rejection of Claims 6, 7, 14 and 15 under 35 U.S.C. § 103 has been overcome.

VI. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: September 27, 2006

Respectfully submitted,

/Wayne P. Bailey/

Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

The Linux filesystem explained

By Mayank Sarup <mayank@freeos.com>
Posted: (2001-01-03 10:08:44 EST by)

The first thing that most new users shifting from Windows will find confusing is navigating the Linux filesystem. The Linux filesystem does things a lot more differently than the Windows filesystem. This article explains the differences and takes you through the layout of the Linux filesystem.

For starters, there is only a single hierarchal directory structure. Everything starts from the root directory, represented by '/', and then expands into sub-directories. Where DOS/Windows had various partitions and then directories under those partitions, Linux places all the partitions under the root directory by 'mounting' them under specific directories. Closest to root under Windows would be c:.

Under Windows, the various partitions are detected at boot and assigned a drive letter. Under Linux, unless you mount a partition or a device, the system does not know of the existence of that partition or device. This might not seem to be the easiest way to provide access to your partitions or devices but it offers great flexibility.

This kind of layout, known as the unified filesystem, does offer several advantages over the approach that Windows uses. Let's take the example of the /usr directory. This directory off the root directory contains most of the system executables. With the Linux filesystem, you can choose to mount it off another partition or even off another machine over the network. The underlying system will not know the difference because /usr appears to be *a local directory that is part of the local directory structure!* How many times have you wished to move around executables and data under Windows, only to run into registry and system errors? Try moving

c:windowssystem
to another partition or drive.

Another point likely to confuse newbies is the use of the frontslash '/'
instead of the backslash '' as in DOS/Windows. So c:windowssystem would
be /c/windows/system. Well, Linux is not going against convention here.
Unix has been around a lot longer than Windows and was the standard a lot
before Windows was. Rather, DOS took the different path, using '/' for
command-line options and '' as the directory separator.

To liven up matters even more, Linux also chooses to be case sensitive.
What this means that the case, whether in capitals or not, of the characters becomes very important. So this is not the same as THIS or This
for that matter. This one feature probably causes the most problems for
newbies.

We now move on to the layout or the directory structure of the Linux
filesystem. Given below is the result of a 'ls -p' in the root directory.

bin/ dev/ home/ lost+found/ proc/ sbin/ usr/
boot/ etc/ lib/ mnt/ root/ tmp/ var/

/sbin - This directory contains all the binaries that are essential to the
working of the system. These include system administration as well as
maintenance and hardware configuration programs. Find lilo, fdisk, init,
ifconfig etc here. These are the essential programs that are required by
all the users. Another directory that contains system binaries is /usr/sbin.
This directory contains other binaries of use to the system administrator.
This is where you will find the network daemons for your system along with
other binaries that only the system administrator has access to, but which are
not required for system maintenance, repair etc.

/bin - In contrast to /sbin, the bin directory contains several useful
commands that are used by both the system administrator as well as
non-privileged users. This directory usually contains the shells like
bash, csh etc. as well as much used commands like cp, mv, rm, cat, ls.
There also is /usr/bin, which contains other user binaries. These binaries
on the other hand are not essential for the user. The binaries

in /bin
however, a user cannot do without.

/boot - This directory contains the system.map file as well as the Linux
kernel. Lilo places the boot sector backups in this directory.

/dev - This is a very interesting directory that highlights one important
characteristic of the Linux filesystem - everything is a file or a
directory. Look through this directory and you should see hda1, hda2 etc,
which represent the various partitions on the first master drive of the
system. /dev/cdrom and /dev/fd0 represent your CDROM drive and your floppy
drive. This may seem strange but it will make sense if you compare the
characteristics of files to that of your hardware. Both can be read from
and written to. Take /dev/dsp, for instance. This file represents your
speaker device. So any data written to this file will be re-directed to
your speaker. Try 'cat /etc/lilo.conf > /dev/dsp' and you should hear some
sound on the speaker. That's the sound of your lilo.conf file! Similarly,
sending data to and reading from /dev/ttyS0 (COM 1) will allow you to
communicate with a device attached there - your modem.

/etc - This directory contains all the configuration files for your system.
Your lilo.conf file lies in this directory as does hosts, resolv.conf and
fstab. Under this directory will be X11 sub-directory which contains the
configuration files for X. More importantly, the /etc/rc.d directory
contains the system startup scripts. This is a good directory to backup
often. It will definitely save you a lot of re-configuration later if you
re-install or lose your current installation.

/home - Linux is a multi-user environment so each user is also assigned a
specific directory which is accessible only to them and the system
administrator. These are the user home directories, which can be found
under /home/username. This directory also contains the user specific
settings for programs like IRC, X etc.

/lib - This contains all the shared libraries that are required by system
programs. Windows equivalent to a shared library would be a DLL file.

/lost+found - Linux should always go through a proper

shutdown. Sometimes
your system might crash or a power failure might take the machine down.
Either way, at the next boot, a lengthy filesystem check using fsck will
be done. Fsck will go through the system and try to recover any corrupt
files that it finds. The result of this recovery operation will be placed
in this directory. The files recovered are not likely to be complete or
make much sense but there always is a chance that something worthwhile is
recovered.

/mnt - This is a generic mount point under which you mount your filesystems
or devices. Mounting is the process by which you make a filesystem
available to the system. After mounting your files will be accessible
under the mount-point. This directory usually contains mount points or
sub-directories where you mount your floppy and your CD. You can also
create additional mount-points here if you want. There is no limitation to
creating a mount-point anywhere on your system but convention says that
you do not litter your file system with mount-points.

/opt - This directory contains all the software and add-on packages that
are not part of the default installation. Generally you will find KDE and
StarOffice here. Again, this directory is not used very often as it's
mostly a standard in Unix installations.

/proc - This is a special directory on your system. We have a more detailed
article on this one here.

/root - We talked about user home directories earlier and well this one is
the home directory of the user root. This is not to be confused with the
system root, which is directory at the highest level in the filesystem.

/tmp - This directory contains mostly files that are required temporarily.
Many programs use this to create lock files and for temporary storage of
data. On some systems, this directory is cleared out at boot or at
shutdown.

/usr - This is one of the most important directories in the system as it
contains all the user binaries. X and its supporting libraries can

be
found here. User programs like telnet, ftp etc are also placed here.
/usr/doc contains useful system documentation. /usr/src/linux contains the
source code for the Linux kernel.

/var - This directory contains spooling data like mail and also the output
from the printer daemon. The system logs are also kept here in /var/log/messages. You will also find the database for BIND in /var/named
and for NIS in /var/yp.

This was a short and basic look at the Linux filesystem. You do need to
have at least this basic knowledge of the layout of the filesystem to
fully utilize its potential. One good place to read about the filesystem
is this detailed document at www.pathname.com/fhs/1.2/fsstnd-toc.html that
specifies the standard structure of the Linux filesystem.

Linux Filesystem Hierarchy Standard
Exploring /proc
Filesystems HOWTO

Other articles by Mayank Sarup

Current Rating: [7.81 / 10] Number of Times Rated: [395]

5 Rate It!